

What is claimed is:

- 1 1. A computer-implemented method of debugging an object-oriented
2 computer program, the method comprising:
 - 3 (a) in response to user input, identifying a plurality of creators for a
4 class defined in the object-oriented computer program and setting a plurality of
5 breakpoints on the identified creators; and
 - 6 (b) halting execution of the object-oriented computer program during
7 debugging in response to hitting any of the plurality of breakpoints.
- 1 2. The method of claim 1, wherein identifying the plurality of creators
2 includes identifying every creator for the class.
- 1 3. The method of claim 1, further comprising, after identifying the plurality of
2 creators, displaying a list of the identified creators and receiving user input to select a
3 subset of identified creators, wherein the plurality of breakpoints are set on only the
4 subset of the identified creators.
- 1 4. The method of claim 1, wherein the plurality of breakpoints are collectively
2 set on all of the identified creators in response to the user input.
- 1 5. The method of claim 1, wherein setting the plurality of breakpoints
2 includes setting each breakpoint from the plurality of breakpoints on a statement in
3 one of the identified creators.
- 1 6. The method of claim 5, wherein setting each breakpoint includes inserting
2 debugging program code in the creator on which such breakpoint is set.
- 1 7. The method of claim 1, wherein identifying the plurality of creators and
2 setting the plurality of breakpoints are performed in response to user input to set a
3 creation breakpoint, and wherein the plurality of breakpoints are associated with the
4 creation breakpoint.

1 8. The method of claim 7, further comprising, in response to the user input to
2 set the creation breakpoint, adding an entry for the creation breakpoint in a breakpoint
3 data structure, wherein setting the plurality of breakpoints includes storing breakpoint
4 information for each breakpoint in the breakpoint data structure, wherein the
5 breakpoint information for each breakpoint is associated with the entry in the
6 breakpoint data structure for the creation breakpoint.

1 9. The method of claim 1, further comprising tracking a total number of hits
2 to the plurality of breakpoints.

1 10. The method of claim 9, wherein halting execution of the object-oriented
2 computer program during debugging in response to hitting any of the plurality of
3 breakpoints includes:

4 (a) determining whether the total number of hits meets a condition in
5 response to hitting any of the plurality of breakpoints; and

6 (b) halting execution of the object-oriented computer program if the
7 total number of hits meets the condition.

1 11. The method of claim 10, wherein the condition is the total number of hits
2 meeting or exceeding a threshold.

1 12. The method of claim 1, wherein each creator comprises a constructor
2 method defined in the class.

1 13. The method of claim 1, further comprising collectively removing the
2 plurality of breakpoints in response to user input.

1 14. A computer-implemented method of debugging an object-oriented
2 computer program, the method comprising:
3 (a) tracking a number of object creations of a class defined in the
4 object-oriented computer program during debugging; and
5 (b) halting execution of the object-oriented computer program in
6 response to the number of object creations meeting a condition.

1 15. The method of claim 14, wherein the condition is the number of object
2 creations meeting or exceeding a threshold.

1 16. The method of claim 14, wherein tracking the number of object creations
2 includes incrementing a counter in response to hitting any of a plurality of breakpoints
3 set on a plurality of creators for the class.

1 17. The method of claim 14, further comprising, in response to user input,
2 identifying the plurality of creators for the class and setting the plurality of
3 breakpoints on the identified creators.

1 18. The method of claim 17, wherein identifying the plurality of creators
2 includes identifying every creator for the class.

1 19. The method of claim 17, further comprising, after identifying the plurality
2 of creators, displaying a list of the identified creators and receiving user input to select
3 a subset of identified creators, wherein the plurality of breakpoints are set on only the
4 subset of the identified creators.

1 20. The method of claim 17, wherein the plurality of breakpoints are
2 collectively set on all of the identified creators in response to the user input.

1 21. The method of claim 17, wherein identifying the plurality of creators and
2 setting the plurality of breakpoints are performed in response to user input to set a

3 creation breakpoint, and wherein the plurality of breakpoints are associated with the
4 creation breakpoint.

1 22. The method of claim 18, wherein each creator comprises a constructor
2 method defined in the class.

11/01/2009 10:00:00 AM

1 23. An apparatus, comprising:

2 (a) a memory within which resides at least a portion of an object-
3 oriented computer program; and

4 (b) program code configured to debug the object-oriented computer
5 program by, in response to user input, identifying a plurality of creators for a
6 class defined in the object-oriented computer program and setting a plurality of
7 breakpoints on the identified creators, and halting execution of the object-
8 oriented computer program during debugging in response to hitting any of the
9 plurality of breakpoints.

1 24. The apparatus of claim 23, wherein the program code is configured to
2 identify the plurality of creators by identifying every creator for the class.

1 25. The apparatus of claim 23, wherein the program code is further configured
2 to, after identifying the plurality of creators, display a list of the identified creators and
3 receive user input to select a subset of identified creators, wherein the plurality of
4 breakpoints are set on only the subset of the identified creators.

1 26. The apparatus of claim 23, wherein the plurality of breakpoints are
2 collectively set on all of the identified creators in response to the user input.

1 27. The apparatus of claim 23, wherein the program code is configured to set
2 the plurality of breakpoints by setting each breakpoint from the plurality of
3 breakpoints on a statement in one of the identified creators.

1 28. The apparatus of claim 23, wherein the program code is configured to
2 identify the plurality of creators and set the plurality of breakpoints in response to user
3 input to set a creation breakpoint, and wherein the plurality of breakpoints are
4 associated with the creation breakpoint.

1 29. The apparatus of claim 28, further comprising a breakpoint data structure,
2 resident in the memory, wherein the program code is configured to, in response to the
3 user input to set the creation breakpoint, add an entry for the creation breakpoint in the
4 breakpoint data structure, and wherein the program code is configured to set the
5 plurality of breakpoints by storing breakpoint information for each breakpoint in the
6 breakpoint data structure, wherein the breakpoint information for each breakpoint is
7 associated with the entry in the breakpoint data structure for the creation breakpoint.

1 30. The apparatus of claim 23, wherein the program code is further configured
2 to track a total number of hits to the plurality of breakpoints.

1 31. The apparatus of claim 30, wherein the program code is configured to
2 determine whether the total number of hits meets a condition in response to hitting
3 any of the plurality of breakpoints, and halt execution of the object-oriented computer
4 program if the total number of hits meets the condition.

1 32. The apparatus of claim 31, wherein the condition is the total number of
2 hits meeting or exceeding a threshold.

1 33. The apparatus of claim 23, wherein the program code is further configured
2 to collectively remove the plurality of breakpoints in response to user input.

1 34. An apparatus, comprising:

2 (a) a memory within which resides at least a portion of an object-
3 oriented computer program; and

4 (b) program code configured to debug the object-oriented computer
5 program by tracking a number of object creations of a class defined in the
6 object-oriented computer program during debugging, and halting execution of
7 the object-oriented computer program in response to the number of object
8 creations meeting a condition.

1 35. The apparatus of claim 34, wherein the condition is the number of object
2 creations meeting or exceeding a threshold.

1 36. The apparatus of claim 34, wherein the program code is configured to
2 track the number of object creations by incrementing a counter in response to hitting
3 any of a plurality of breakpoints set on a plurality of creators for the class, and
4 wherein the program code is further configured to, in response to user input, identify
5 the plurality of creators for the class and set the plurality of breakpoints on the
6 identified creators.

1 37. The apparatus of claim 36, wherein the program code is configured to
2 identify the plurality of creators and set the plurality of breakpoints in response to user
3 input to set a creation breakpoint, and wherein the plurality of breakpoints are
4 associated with the creation breakpoint.

1 38. A program product, comprising:

2 (a) program code configured to debug an object-oriented computer
3 program by, in response to user input, identifying a plurality of creators for a
4 class defined in the object-oriented computer program and setting a plurality of
5 breakpoints on the identified creators, and halting execution of the object-
6 oriented computer program during debugging in response to hitting any of the
7 plurality of breakpoints; and

8 (b) a signal bearing medium bearing the program code.

1 39. The program product of claim 38, wherein the signal bearing medium

2 includes at least one of a transmission medium and a recordable medium.

IBM CORPORATION
ARMONK, NEW YORK 10504
ATTORNEYS
FLETCHER SMITH & ASSOCIATES
NEW YORK, NEW YORK

1 40. A program product, comprising:

2 (a) program code configured to debug an object-oriented computer
3 program by tracking a number of object creations of a class defined in the
4 object-oriented computer program during debugging, and halting execution of
5 the object-oriented computer program in response to the number of object
6 creations meeting a condition; and

7 (b) a signal bearing medium bearing the program code.